ハッシュ関数の構成法に関する一考察 (言語,代数系および計算機システム)

AUTHOR(S):

井上, 徹

CITATION:

井上, 徹. ハッシュ関数の構成法に関する一考察 (言語,代数系および計算機システム). 数理解析研究所講究録 1999, 1106: 66-80

ISSUE DATE:

1999-07

URL:

http://hdl.handle.net/2433/63256

数理解析研究所講究録
1106巻 1999年 66-80

ハッシュ関数の構成法に関する一考察

ＡＭＳＬ[+]　井上徹、　Toru INOUE

[+]（高度移動通信セキュリティ技術研究所）

現在使われているハッシュ関数にはＳＨＡ，ＳＨＡ－１，ＭＤ４，ＭＤ５などがある。これらのハッシュ関数用の特定のアルゴリズムを用いる方式には演算スピードはそれなりに高速であるが、いくつかのアタック（攻撃）が発表されている方式もある。[1) 2) 3)] 一方、ブロック暗号を圧縮関数として繰り返し用いてハッシュ関数を得る方法に Davies-Meyer の方法がある。[4) 5)] 安全性改善策として２次元 Cellular Automaton を使う方法も報告されている。[6)] また種々の解析結果も報告されている。[7)] しかし一般の６４ビットブロック暗号をそのまま用いるとバースデーアタックにより平均2^{32}の試行で衝突（collision）が生じてしまい、これは今日では安全性が高いとは言えない。ブロック暗号をハッシュ関数に用いる利点としてブロック暗号の安全性が保証されていればそれを用いたハッシュ関数も安全性が保証されると言われている。[8) 9)] また、単一のハッシュモードを多重の（multiple）ハッシュモードに拡張しても安全性は保たれる。[9)] さらにデータの暗号化部で使う暗号アルゴリズムのハードウェアまたはソフトウェア資源が共用で

きるメリットがある。本稿ではKnudsenらの方法を大幅に改良し、効率の良いハッシュ値を得る方法を提案する。

１．はじめに

本稿では安全性が保証されたブロック暗号を用いることとし、以下の仮定をおくことにする。

１）　単一のハッシュモードは安全（secure）である。

２）　単一のハッシュモードを解読(break)する短絡（short cut）が存在しない。

３）　ブロック暗号の安全性が保証されていればそれを用いたハッシュ関数も安全性が保証される。

４）　文書は十分に長いとする。

しかしながら、一般の６４ビット暗号をそのまま用いたのでは衝突（バースデーアタックまたは collision ）が平均2^{32}回の試行で生じ、これは今日の暗号事情ではけっして安全性が高いとはいえない。[9)]

誤り訂正符号を用いて衝突耐性を向上させる手法にKnudsen L. and Preneel B.の手法がある。[8) 9)] 彼らの手法は４元（ｎ，ｋ，ｄ）線形符号を用いて衝突を$2^{m/2}$から$2^{(d-1)m/2}$に改善している。しかし彼らは非２元（non-binary）符号しか構成に用いておらず、安全性の面でもハードウェア構成の面でも改良の余地がある。筆者はＢＣＨ符号のような２元符号を用いた構成法を試み，その有用性を示す。更に畳み込み符号を用いた手法も考察し、ブロック符号と同じ性能条件で設計するとき、拘束長Ｎの畳み込み符号

はブロック符号を用いた時の１／Ｎの数の Davies-Meyer 関数で装置化できることを示す。

2.ハッシュ関数

任意の長さの文書をある決められた長さに圧縮するためには暗号論的ハッシュ関数がもちいられる。衝突とはｘ≠ｘ′なる場合にｈ（ｘ）＝ｈ（ｘ′）となることである。衝突に対するハッシュ関数の耐性の定義には、弱い耐性と強い耐性がある。弱い耐性（Weak collision resist）とはプレイメージ耐性(pre-image resistance)あるいはセカンドプレイメージ耐性（2nd-preimage resistance）と呼ばれるものである。

プレイメージ耐性とは与えられたハッシュ値Ｈ＝ｈ（ｘ）に対してｈ（ｘ′）なるｘ′を見つけることがどのくらい困難かということである。

セカンドプレイメージ耐性とは与えられた文書入力ｘに対してハッシュ値Ｈ＝ｈ（ｘ）＝ｈ（ｘ′）を持つ第２の文書入力ｘ′を見つけることがどの程度困難かということである。つまり、文書ｘとそのハッシュ値Ｈ＝ｈ（ｘ）があったとき、同じハッシュ値Ｈ＝ｈ（ｘ）になる別の文書ｘ′を見つけることがいかなる文書ｘに対しても困難であり、平均Ｗ回の試行を必要とすればそのハッシュ関数ｈのセカンドプレイメージ耐性はＷであるという。本稿では以後プレイメージとセカンドプレイメージとは数値として同じであるので同等に扱い区別しない。

強い耐性(strong collision resist)とは、衝突耐性(collison resistance)またはバースデーアタックに対する耐性と呼ばれるものである。ｈ（ｘ）＝

h（x′）となるいかなる文書入力ペアー（x、x′；x≠x′）を見つけることがどの程度困難かということである。すなわち、ハッシュ値が同じになる異なる文書の組を見つけるために、平均W回の試行を必要とするならば、そのハッシュ関数のバースデーアタックに対する耐性はWであるという。通常単に衝突耐性と言うときはこの強い耐性を言う。

m—ビットブロック暗号（ＤＥＳなどは６４－ビットブロック暗号である）に基くハッシュ関数のハッシュレート（hash rate）とは、１回の暗号化または復号化で処理されるm—ビットメッセージブロックの数で定義される。E_k（・）をm—ビットブロック暗号の暗号アルゴリズムとし、そのm—ビット鍵をＫとする。ＦＥＡＬのように６４ビットブロック暗号で６４ビット鍵を用いる暗号が該当する。圧縮関数ｈを Davies-Meyer 関数と呼ぶ。

『**仮定**１：関数ｈに対するコリジョン（collision）を見つけるには安全なブロック暗号を用いている限り（m—ビットブロックの）約$2^{m/2}$回の暗号化が、又関数ｈに対するプレイメージ（preimage）を見つけるには約2^m回の暗号化が必要である。』[8) 9)]

メッセージM_i、ハッシュ（hash）値H_iおよびひとつ前の（previous）ハッシュ値H_{i-1}の間には

$$H_i = h(M_i, H_{i-1}) = E_{Mi}(H_{i-1}) + H_{i-1}$$

が成り立つ。ここで＋は法２加算である。H_iは文書の始まりから時刻ｉ—１までのハシュ値と時刻ｉのメッセージとの累積加算値である。

$E_{K'}$（・）を、ａ＞０なるａ×m—ビット鍵Ｋを使うm—ビットブロック暗

号アルゴリズムとする。h_1、h_2、…、h_nを鍵を互いに異なる値を取ることにより互いに異なる Davies-Meyer 関数にする。多重デヴィスマイヤー（multiple Davies-Meyer）関数はｒ個のｍ—ビットメッセージ入力をアフィン変換してｎ個のペアー（X_i，Y_i）へ写像して入力する。出力はH_1、H_2、…、H_nの連接となる。コリジョン（collision）またはプレイメージ(preimage)アタックにおいて入力ブロックを構成する（X_i，Y_i）が元のペアーと異なるならｈ（X_i，Y_i）は積極的（active）と呼ばれる。又、２つの関数h_i（X_i，Y_i）とh_j（X_j，Y_j）は独立に（independently）攻撃可能という時は、関数h_iの変数パラメータ（X_i，Y_i）が変わっても、関数h_jの変数パラメータ（X_j，Y_j）は変わらないような関数をいう。

仮定２：多重デヴィスマイヤー（multiple Davies-Meyer）の圧縮関数に対するコリジョン（collision）あるいはプレイメージ（preimage）が見つかったとする。Ｐを active な関数の数とし、Ｐ－ｖを独立に攻撃可能な関数の最大数とする。このコリジョン（collision）あるいはプレイメージ（preimage）が起こるためには少なくとも$2^{vm/2}$，あるいは2^{vm}回の暗号化がそれぞれ必要である。』[8) 9)]

3.誤り訂正符号を用いた構成法

3.1. Knudsen L, and Preneel Ｂによる構成法

Knudsen L, and Preneel Ｂによる誤り訂正符号を用いた構成法を紹介しよう。

［定理３］：長さｎ、情報シンボル数ｋ、最小距離ｄ、（ａ＋１）ｋ＞ｎ但

しａ≧１且つｍ＞＞ｌｏｇ$_2$ｎなるＧＦ（２$^{a+1}$）上の（ｎ、ｋ、ｄ）符号で入力ブロックを符号化したとする。すると仮定２が成り立つ限り圧縮関数に対するコリジョン（collision）を見つけることは少なくとも２$^{(d-1)m/2}$回の暗号化が、あるいはプレイメージ（preimage）を見つけるには少なくとも２$^{(d-1)m}$回の暗号化が必要である。[9)]』

（証明）：入力は（ａ＋１）ｋ個のｍビットブロックからなる。ｎ個の連鎖値Ｈ$_{i-1}^{1}$からＨ$_{i-1}^{n}$までとｒ個のメッセージブロックＭ^{1}からＭ^{r}までのブロックからなる。ここでｒ＝２ｋ－ｎ＞０である。入力ブロックの（ａ＋１）個の引き続くビットはＧＦ（２$^{a+1}$）のｋ個の要素を与えるように構成される。

これら要素は（ｎ，ｋ，ｄ）符号を使いＧＦ（２$^{a+1}$）のｎ個の元となる。ｎ個の関数の１つに対して（ａ＋１）ビットの入力を表す。個々のビットはＧＦ（２$^{a+1}$）の上のベクトル空間としてＧＦ（２$^{a+1}$）の元を表している。

この構成はｖ＝ｄ－１なる値に対して仮定２に対する条件が満足される。（ａ＋１）ｋ個の入力の線形変換によって圧縮関数を得ることが可能となる。ここで最初のｋ個の関数がお互いに独立である。符号語は最後のｎ－ｋ個の関数の少なくともｄ－１個が最初の関数の入力に依存している。ｎ－ｋ≧ｄ－１だから定理３はおのずと証明される。（証明終）

この hash 関数はｎ・ｍビットの内部メモリーが必要で hash rate は（ａ＋１）（ｋ／ｎ）－１である。すなわち、例えば距離３の誤り訂正符号を使う

ことによってコリジョン（collision）を$2^{m/2}$から2^mに、あるいはプレイメージ（preimage）を2^mから2^{2m}に改善し、安全な hash 関数を簡単に構成できる。

3.2. ブロック符号による構成法

また新たにＢＣＨ符号などの２元符号を用いる方式[10]が提案され、１つのｍビット入力に２元符号（binary code）の２つの符号語を割り当てる構成にして２元符号を使えるような構成にしている。

4. 提案方式

前項（３．２．項）の改善策はガロア体の符号を２元の符号に置き換えてはいるものの、初期値を除くと１単位時間前のハッシュ値だけで構成されるｎ個のｍ－ブロックから構成されている１列目のＢＣＨ符号語は前の時刻のハッシュ値だけを符号化する事になり、ここには新しいメッセージの情報は入力されない。従って、この列が攻撃（アタック）されることはなく、攻撃に対する符号化をする必要もない。符号化は２列目のｎ個のｍ－ブロックだけを行い攻撃から守ればよい。ベクトルX_iは暗号器が完全（乱数オラクルとみなせる）であれば乱数と見做せる。関数h_iのY_iだけ誤り訂正符号化する。

このように構成しても圧縮関数に対するコリジョン（collision）を見つけることは少なくとも$2^{(d-1)m/2}$回の暗号化が、あるいはプレイメージ（preimage）を見つけるには少なくとも$2^{(d-1)m}$回の暗号化が必要である。

4.1.ブロック符号を用いた方法

［定理４］：ｋ個のメッセージブロックをＥＣＣ符号化してｎ個のｍ—ビットブロックを作り、それを n 個の previous hashed values と並べて、ｎ×２個のｍ—ビットブロックにしてｎ－multiple Davies-Meyer 関数に入力するとする。n 個のブロックに collision または preimage が起こったとしＰ個の Davies-Meyer 関数が積極的ならばＰ－（ｄ—１）個のメッセージ入力ベクトルY_iが独立でｄ—１個のメッセージ入力ベクトルY_iが従属である。』

（証明）：ベクトルY_iのｎビットは少なくとも最初のｋビットが互いに独立である。最後のｎ－ｋビットの少なくともｄ－１個が最初の入力に依存している。ｎ－ｋ≧ｄ－１だから定理４はおのずと証明される。(証明終)

以上の議論は定常状態で入力ベクトルX_iが乱数オラクルの出力とみなせる限り成り立つ。しかし、初期状態においては鍵入力、メッセージ入力ともに任意の値を取りうる。従って、初期入力ベクトルについては Knudsen, L. and Preneel, B.に基づいて入力せねばならない。

［仮定２の拡大（extension)］：初期状態： Multiple-Davies-Meyer の圧縮関数に対するコリジョンあるいはプレイメージが見つかったとする。Ｐを active な関数の数、Ｐ－ｖを独立に攻撃可能な関数の数とする。f_1, …, f_nを同時に衝突が起こっている Davies-Meyer 関数とする。個々の関数には

$$f_i(X_i, Y_i) = E_{X_i}(Y_i) + Y_i$$

が成り立つ。これらは異なる値を$\lceil \log_2 n \rceil$個の鍵を固定して得られる。２ｋ個のｍビット初期値入力をもつ圧縮関数を考えそれがアフィン変換によりｎ個のペアー（X_i，Y_i）に写像されるすべての圧縮関数の出力が２ｋ個のブロックに依存する。すなわち変換マトリクスは階数２ｋを持つ。

定常状態：ｋ個のｍビット入力を持つ初期値状態を除く定常状態においてｋ個のｍビット入力を持つ圧縮関数のアフィン変換によりｎ個のベクトルY_iに写像されるすべての圧縮関数の出力がｋ個のブロックに依存する。すなわち変換マトリクスは階数ｋを持つ。

同時衝突がf_1，…，f_nに起こった時、すなわち２つの入力値$\{Z_i\}$と$\{Z_i'\}$の異なった組がすべてのｎ個の関数に等しい出力を与えるとする。この時$P \leqq n$で$Z_i \neq Z_i'$に依存するＰ個の関数の組を$\{f_j\}$とする。関数$\{f_j\}$のマトリクスの階数はＰ－ｖとなる。これらＰ個の関数$\{f_j\}$に対する同時衝突は少なくとも$2^{v m/2}$回の暗号化が、また、pre-image に対しては$2^{v m}$回の暗号化が必要である。従って鍵入力ｍビットのうちｋビットを予め誤り訂正符号で符号化しｎビットにして初期値X_0を与え、残りｎ－ｋビットを初期値Y_0のうちｎ－ｋビットに配置する。すなわち初期状態のメッセージビットはｒ＝２ｋ－ｎビットとなる。２番目以降のメッセージビットはｒ＝ｋビットとなる。

すなわちｄ—１個のY_iベクトルだけが従属であることが要求されるので定常状態においてはベクトルY_jだけを誤り訂正符号化すればよい。

［補題５］：符号長ｎ、情報記号数ｋ、最小距離ｄの２元符号を考える。デ

ビスマイヤー（Davies-Meyer）の段数をｎとする。時刻をｉと仮定すると前の時刻ｉ－１のｎ個のハッシュ値（previous hash value）はすべて１列目のｎ個のｍ－ブロックに戻されるから、メッセージブロックのため使われるブロック数はｋとなる。Hash rate（ハッシュ率）はｋ／ｎとなる。』

これは元々誤り訂正符号の持っている符号化率に等しい。すなわち効率が大幅に改善されることになる。運用上の注意としては初期値にｎ×ｍビットの鍵の初期値が必要である。

新しい構成法は２元符号の１つの符号語によってｎ個のペアー入力の片側であるｍビットブロックに入力される。

２元符号を使った例とKnudsenらの結果とを比較のため表でしめす。

表１．　２元の符号による方法と従来法との比較

field	t	code	rate	collision	memory
GF(2)	1	(7, 4, 3)	4/7=0.571	2^m	7m
GF(2)	1	(15,11,3)	11/15=0.733	2^m	15m
GF(2)	2	(15,7,5)	7/15=0.467	2^{2m}	15m
GF(2)	2	(25,15,5)	15/25 =0.60	2^{2m}	25m
GF(2)	2	(30,20,5)	20/30 =0.666	2^{2m}	30m
	2	(62,54,5)	54/62 =0.871	2^{2m}	62m
$GF(2^2)$	1	(5,3,3)	1/5=0.20	2^m	5m
$GF(2^4)$	1	(6,4,3)	1/4=0.25	2^m	6m
$GF(2^2)$	1	(8,5,3)	1/4=0.25	2^m	8m

Non-binary の符号は **Knudsen L. and Preneel B.** [8) 9)] らの結果による。

この方法は **Knudsen L. and Preneel B.**と比べてフィードバックされたハッシュ値を誤り訂正符号化しない分だけ効率（hash rate）が改善されており、ＢＣＨ符号などの２元符号を使うため既存の高速ハードウェアなどが流用で

きる。更に collision をｍ＝６４として２^{2m}＝２^{128}に一回の割合に抑える要望があればｄ＝５のＢＣＨ符号を選べば良い。

以上の説明からわかるように本稿によれば２元符号を使うため、ガロア体の演算をしなくてすみ、且つ、より安全性の高いハッシュ値を得ることができる。

4.2. 畳み込み符号を用いた構成法[11)]

ブロック符号では誤り訂正能力を大きくするには符号長が一般に大きくなり演算が複雑になる傾向があった。又、符号長分だけのデビスマイヤーの段数が必要になるため多くのデビスマイヤー関数を装置化しておく必要がある。

本稿ではデビスマイヤー（Davies-Meyer）の基本構成である段数入力を畳み込み符号の小ブロック長n_0に選ぶことにより又、拘束長Ｎとするときｎ段に分けて入力することによりデビスマイヤーのサイズをブロック符号を使った時の符号長ｎから畳込み符号の小ブロック長n_0まで小さくできる構成法を示す。すなわち、デビスマイヤー（Davies-Meyer）関数の基本構成の段数は１／Ｎまで減少する。

構成は畳み込み符号器と多段デビスマイヤーハッシュ関数とＮ段に連接されたハッシュ値を蓄積するＦＩＦＯメモリーによって特徴付けられる。畳み込み符号は大きく分けて２種類の符号器がある。例で説明しよう。畳み込み符号の小ブロック長n_0ビット、小情報記号ビットk_0、拘束長Ｎ段とするとき（n_0，k_0，Ｎ）畳み込み符号と呼ぶ。定理４を畳み込み符号にも適用しよう。

[定理６]：n_0個の previous hashed values とk_0個のメッセージブロックを畳み込み符号化してn_0個のｍ—ビットブロックを作り$n_0 \times 2$個のｍ—ビットブロックをn_0－multiple Davies-Meyer 関数に入力するとする。拘束長Ｎとするとき$N \times n_0$個のブロックに collision または preimage が起こったとしＰ個の Davies-Meyer 関数が積極的ならばＰ－ｄ＋１個の入力ベクトルが独立でｄ—１個の入力ベクトルが従属である。』

運用上の注意としては$n_0 \times$ｍビットの鍵初期値が必要である。

畳込み符号を用いると多重デビスマイヤー関数のサイズは１／Ｎ倍と小さくなるが、拘束長Ｎ，全長$N \times n_0 \times$ｍビットのハッシュ値を得るためにはデータの最後に（Ｎ－１）$\times k_0 \times$ｍビットの０を入力してデータを完全に符号器から出力させる必要がある。したがって符号化のため（Ｎ－１）×ｍビットの遅延が起こる。

提案方式による一般的なハッシュ関数は多重デビスマイヤー関数の入力側で$n_0 \times 2$の２次元配置を構成して第１の列のn_0のｍ－ビットブロックに１単位時刻前のn_0個のハッシュ値をフィードバックし新しいメッセージk_0個のｍ－ビットブロックを符号化してn_0個のｍ－ビットブロックにしたのち、入力する方式となる。

ワイナーアッシュ符号[13)]を使った場合とＣＳＯＣ符号（Convolutional Self-Orthogonal Codes；自己直交畳込み符号）[13)]を使った例を表２，３に示す。

表２　ワイナーアッシュ（Wyner-Ash）符号による構成例

符号 (W-S)	N	Hash rate	collision	Memory(Hash)
(4、3、3)	2	3/4	2^m	8 m
(8、7、3)	2	7/8	2^m	1 6 m

表３　ＣＳＯＣ符号による構成例

符号 (CSOC)	N	Hash rate	collision	Memory(Hash)
(3、2、3)	3	2/3	2^m	9 m
(3、2、5)	1 4	2/3	2^{2m}	4 2 m

５．まとめ

誤り訂正符号を用いてハッシュ関数を構成する Knudsen L. and Preneel B. を改良して

２元の符号を用いる方法、

畳み込み符号を用いる方法

を開発し、その構成を示した。近年ブロック暗号の鍵長の安全性が真剣に議論されるようになり、１２８ビット以上の鍵を持つ暗号が主流となりつつある。[14] このような要請と合わせ既存の手法を組み合わせて安全なハッシュ関数を求める研究がもっと活発になされるべきである。

参考文献

1)Dobebertin H., "Cryptanarysis of MD4,: Fast Software Encryption, LNCS 1039, D. Goldman, Ed., Spring-Verlag, 1996, pp.53-69.

2)Dobebertin H., :Cryptanarysis of MD5 compress, Presented at the rump session

of EUROCRYPT'96, May 1996.

3)Kuwakado Hidenori and Tanaka Hatsukazu," On the One-wayness of the reduced MD4 compression function", The 1999 Symposium on Cryptography and Information Security", pp.247-250. Kobe, Japan, January 26-29, 1999.

4)Davies D.W. and Price W.L.,: Digital Signature-An Update,: Proceedings of International Conference on Computer Communications, Sydney, Oct 1984, North Holland: Elsevier, 1985, pp.843-847.

5)Matyas S.M., Meyer C. H. and Oseas J.,: Generating Strong One-Way Functions with Cryptgraphic Algorithm, IBM Technical Disclosure Bulletin, v. 27, n. 10A, Mar. 1985, pp.5658-5659.

6)Hirose Shouichi and Yoshida Susumu," A one-way hash function based on a two-dimensional cellular automaton", The 20-th Symposium on Information Theory and Its Applications (SITA97) pp.213-216, Matsuyama, Japan, December 2-5, 1997.(In Japanese)

7)Morita Hikaru, Odagi Hideo and Ohta Kazuo," Collision search of a hash function by using random mapping", IEICE Trans. Fundamentals, vol. E81-A,No.1, pp.35-40, January 1998.

8) Knudsen L.R.,and Preneel B.,: Hash functions based on block ciphers and quaternary codes, Advances in Cryptology, Proc. Asiacrypto'96, LNCS 1163, K. Kim, T. Matsumoto, Eds., Springer-Verlag, 1996, pp.77-90

9)Knudsen L. and Preneel B. "Fast and secure hashing based on codes,

Crypto'97, LNCS1294、 pp.485-498, 1997.

10)INOUE Toru and TANAKA Hatsukazu, :A Note on Hash Functions Based on Block Ciphers and Error Correction Code, International Symposium on Information Theory and its Applications, pp.127-129., Mexico City MEXICO, October 14-16,1998

11)井上徹、"畳み込み符号を用いたハッシュ関数の一構成法"、信学技報ＩＴ９８－５５，pp.1-6,(1999-01)。

12)Lin, S,: An Introduction to Error Correction Codes, Englewood Cliffs., N., J., 1970,Chapter 10.

13)川村信一：ＡＥＳ参加報告、信学会技術報告、ISEC98-41,pp.39-43, November 1998.